




Detección de Fake News

Inteligencia Artificial II. Facultad de Ingeniería, UNCuyo



La propagación de noticias falsas es un desafío en el mundo digital que afecta la confianza y la toma de decisiones.

Objetivo

Desarrollar y comparar modelos supervisados de clasificación para detectar noticias falsas.





Dataset

Fuente: Kaggle (Fake News Detection).

Cantidad de registros: 44.898.

	title	text	label
0	Ben Stein Calls Out 9th Circuit Court: Committ...	21st Century Wire says Ben Stein, reputable pr...	fake
1	Trump drops Steve Bannon from National Securit...	WASHINGTON (Reuters) - U.S. President Donald T...	true
2	Puerto Rico expects U.S. to lift Jones Act shi...	(Reuters) - Puerto Rico Governor Ricardo Rosse...	true
3	OOPS: Trump Just Accidentally Confirmed He Le...	On Monday, Donald Trump once again embarrassed...	fake
4	Donald Trump heads for Scotland to reopen a go...	GLASGOW, Scotland (Reuters) - Most U.S. presid...	true

Preprocesamiento

	text	label
0	21st century wire says ben stein reputable pro...	0
1	washington reuters us president donald trump r...	1
2	reuters puerto rico governor ricardo rossello ...	1
3	on monday donald trump once again embarrassed ...	0
4	glasgow scotland reuters most us presidential ...	1

1 Eliminación de duplicados

(Filas eliminadas: 6252)

2 Verificar balanceo de clases

(de etiquetas y palabras)

3 Conversión a minúsculas y eliminación de caracteres no alfanuméricos



Implementación Técnica

Conversión de datos:

- BoW para modelos básicos.
- Tokenización para Transformers (entero basado en frecuencia)

División de datos:

- 80% entrenamiento, 20% prueba.

Bibliotecas usadas: Scikit-learn, Keras, Pandas.

Regresión Logística

Modelo lineal rápido y eficiente.





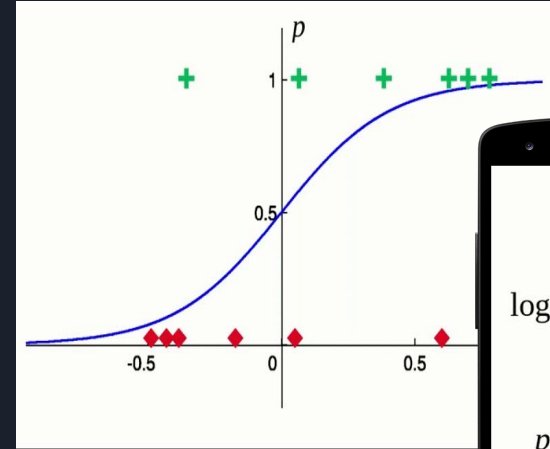
Ajusta pesos para cada palabra, identificando los más relevantes (minimizando función de pérdida)

Calcula la probabilidad de pertenencia a una clase usando la función **sigmoide**.

Simple, rápido y eficiente para grandes volúmenes de texto.

Dificultades con datos **no lineales** o relaciones complejas.

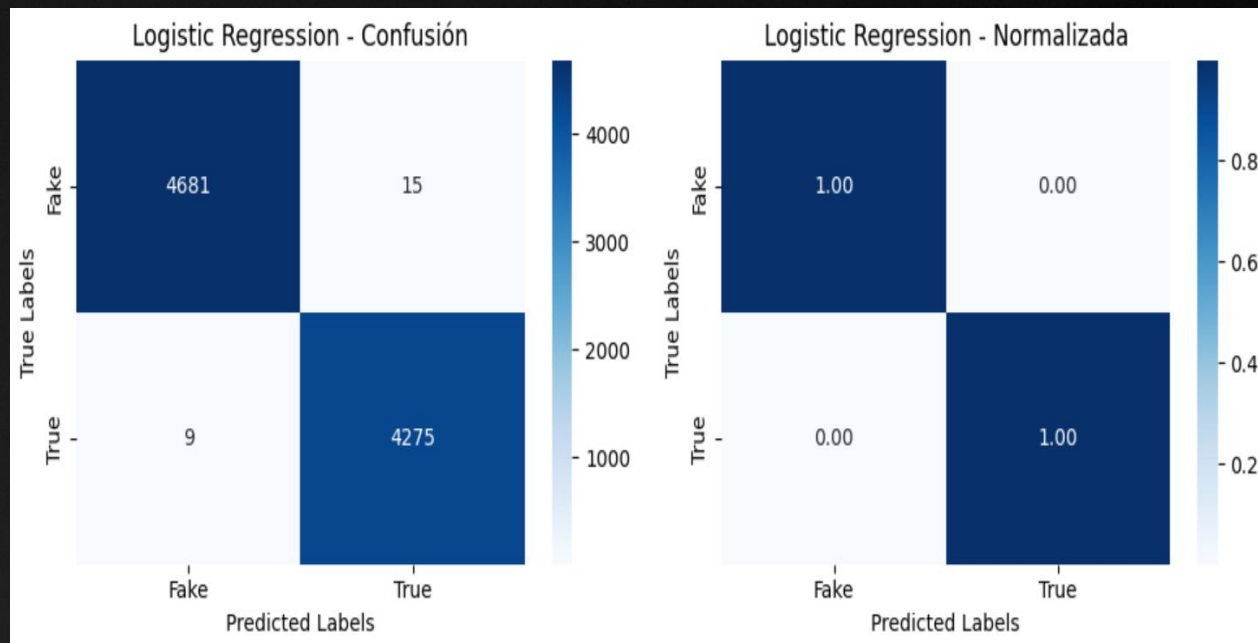
- Número de **Términos** = Tamaño del vocabulario (n° palabras únicas en BoW) + b = **203.677**
- **Iteraciones** realizadas durante el entrenamiento: **92**
- Tipo de **regularización** usado: **L2**



$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x$$

↓

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$




Accuracy - Logistic Regression: 0.997

Naive Bayes

Clasificación probabilística





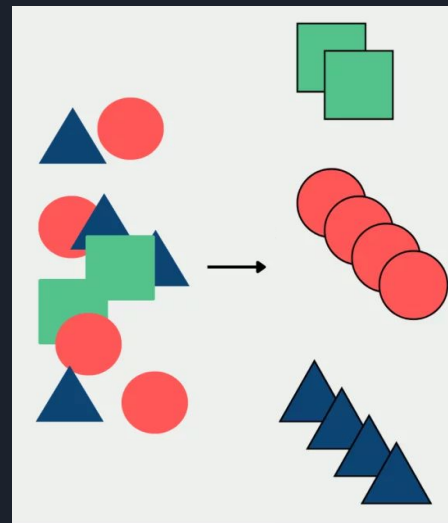
Basado en el
Teorema de Bayes.

Cada palabra
contribuye
independientemente a
la probabilidad total de
la noticia ser "falsa" o
"real".

Rápido y efectivo con
datos textuales
grandes.

La suposición de
independencia entre
palabras **no siempre**
es realista.

- Conteo total de palabras por clase:
18785 y 17133
- El modelo prioriza palabras más
comunes y descarta raras.



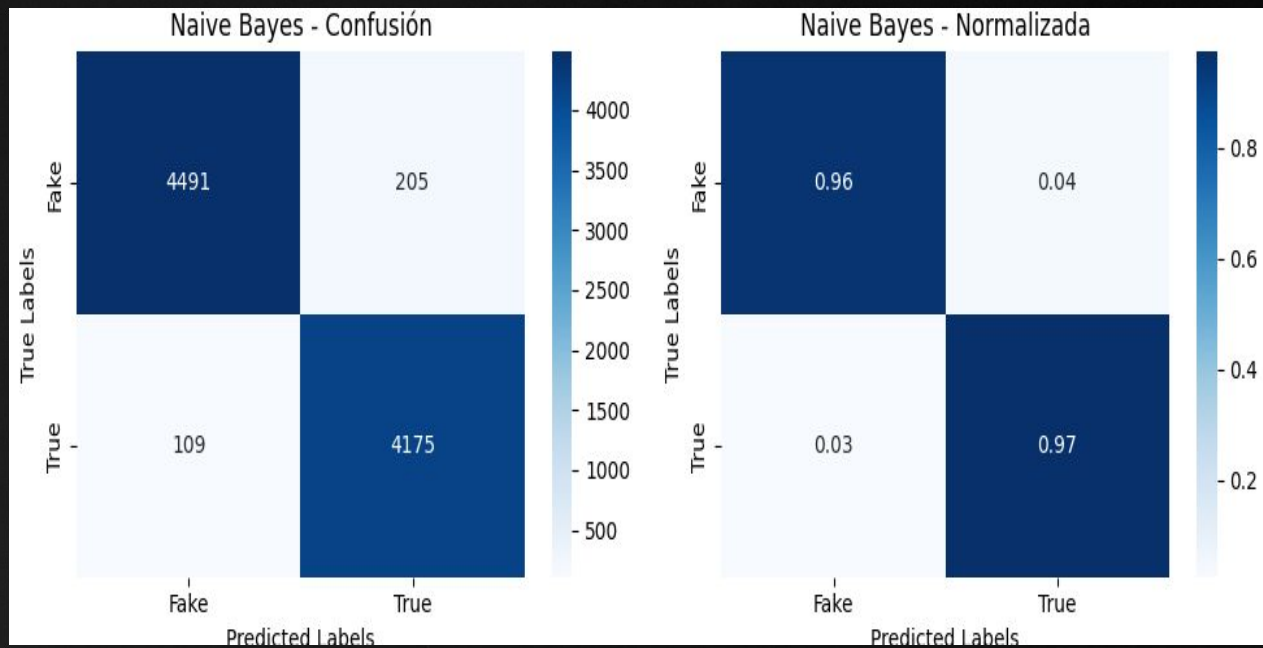
$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

$P(A)$: Probabilidad de A

$P(R|A)$: Probabilidad de que se de R dado A

$P(R)$: Probabilidad de R

$P(A|R)$: Probabilidad posterior de que se de




Accuracy - Naive Bayes: 0.965

K-Nearest Neighbors

Basado en proximidad





Evalúa similitudes
entre textos en
función de las
palabras que
comparten.

Calcula la distancia
entre el texto a clasificar
y los demás.

Toma los **K vecinos** más
ceranos.

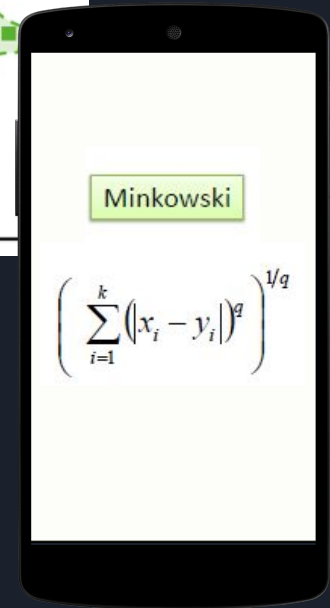
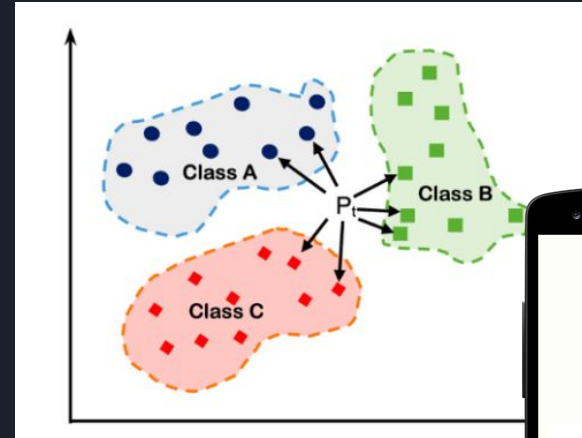
Clasifica según la **clase**
más frecuente entre
ellos.

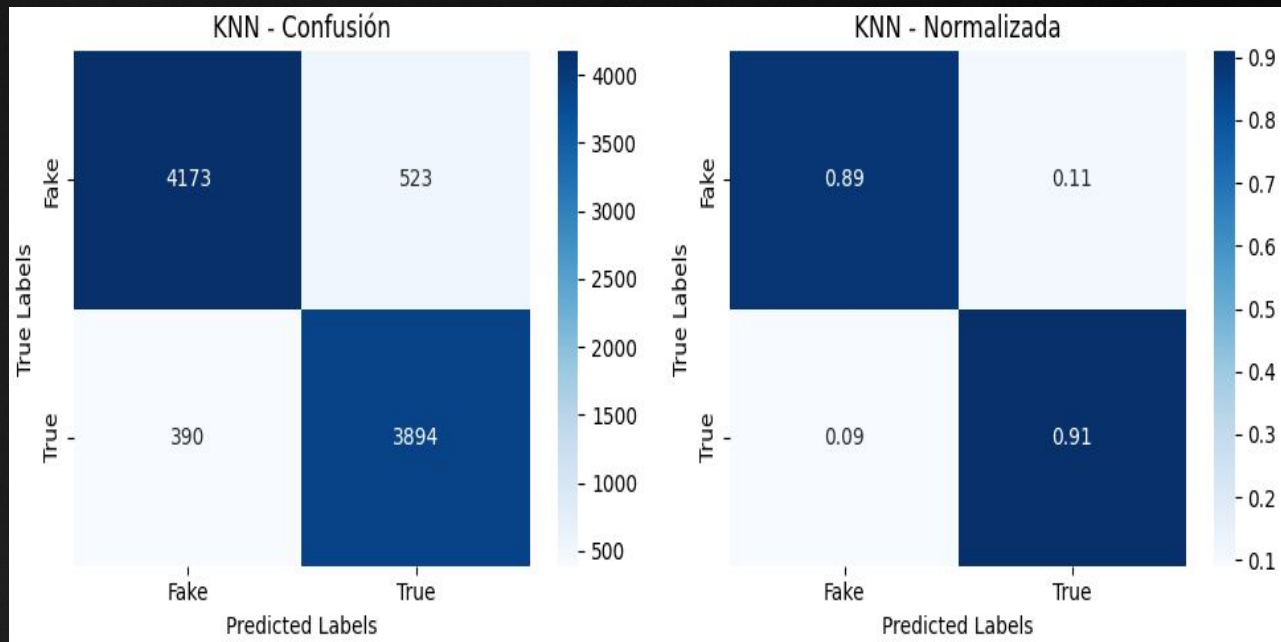
Simple y no necesita
entrenamiento.

Lento para grandes
volúmenes de datos.

Sensible al ruido y a
alta dimensionalidad

- Vecinos considerar = 5
- weights='uniform'. Todos los vecinos tienen **igual peso** en la votación.
- Métrica utilizada para la distancia: **minkowski**

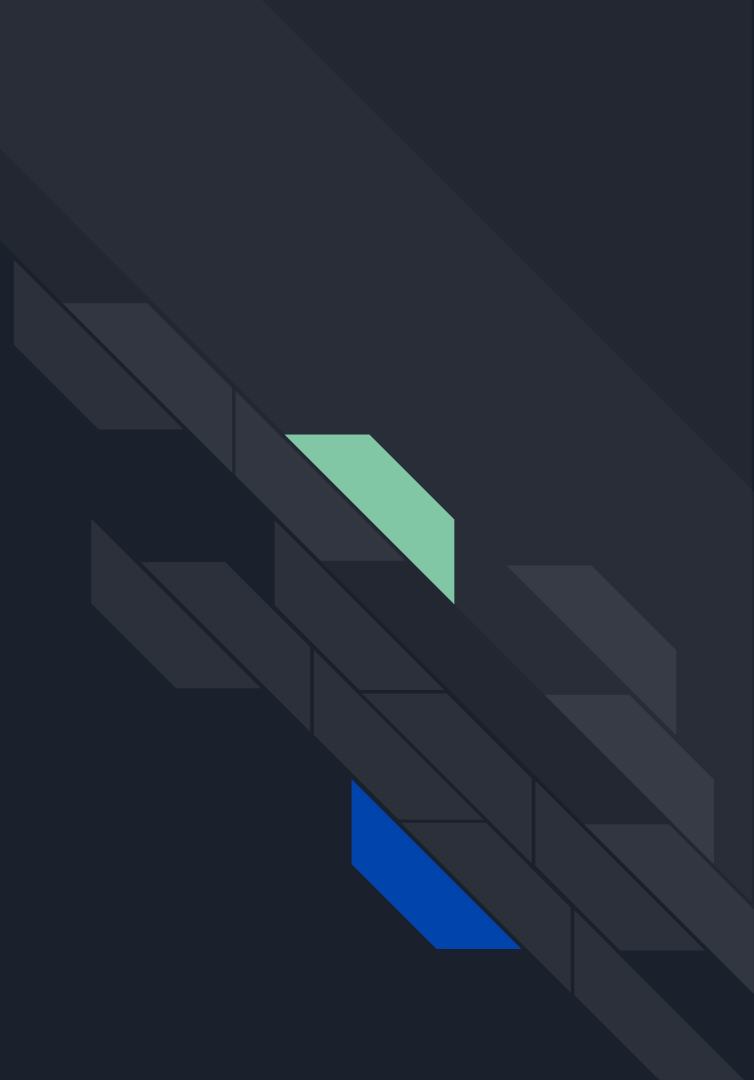




Accuracy - KNN: 0.898

Transformers

Red neuronal con mecanismo de atención





Entrada: Texto tokenizado.

Embedding: palabra a vector.

Transformer: Captura relaciones entre palabras, incluso si están lejos.

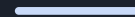
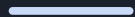
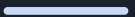
Capa Salida: Clasifica

Mecanismo de **atención** para analizar la relación entre **todas** las **palabras** de un texto **simultáneamente**

Captura **contexto** y relaciones complejas.

Maneja **secuencias largas**.

Alto costo y tiempo.



Entrada

Secuencia de longitud 200. Cada palabra es un entero que corresponde a su posición en el vocabulario.

Capa de Embeddings

TokenAndPositionEmbedding

Bloque Transformer

TransformerBlock

**Capa de Pooling global
(GlobalAveragePooling1D)**

Reduce la dimensionalidad de la secuencia



Dropout (30%)

Después del pooling para estabilizar las representaciones aprendidas antes de las capas densas.

Capa Densa

10 neuronas y activación ReLU.
Salida: vector de tamaño (batch_size, 10).

Dropout (30%)

Regularización antes de la salida.
Para prevenir sobreajuste.

Capa Densa (1 neurona, Sigmoid)

1 neurona y activación sigmoide.
Salida: valor entre 0 y 1 (probabilidades)

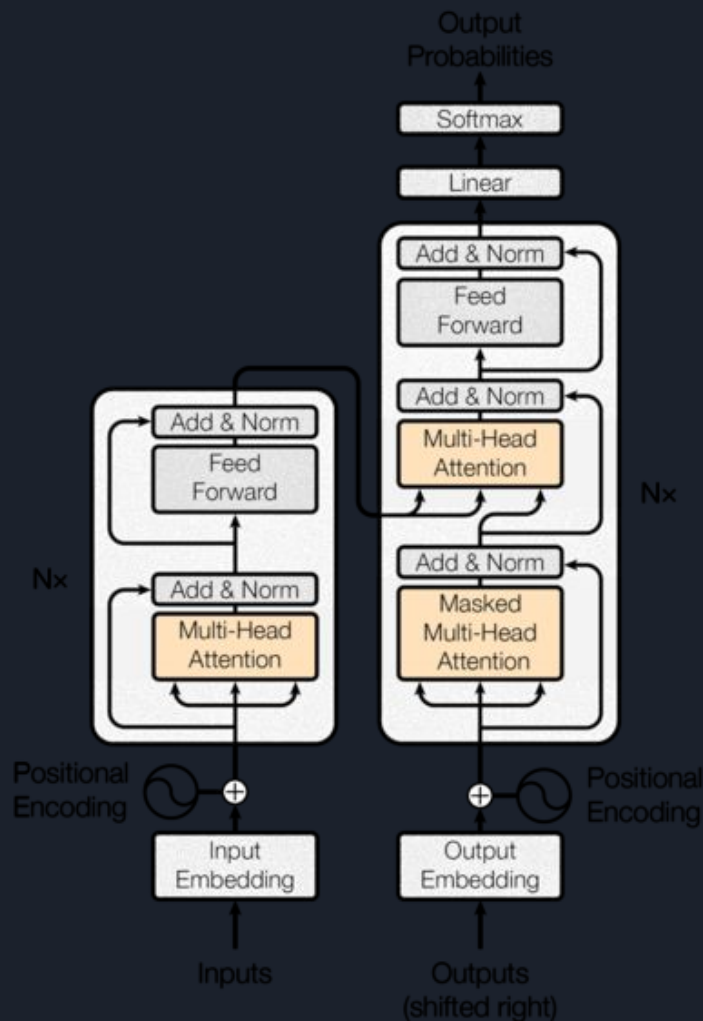
Bloque Transformer

Atención Multi-Cabezal (2): permite el enfoque diferentes partes simultáneamente.

Red Feed-Forward: una con activación ReLU y otra con dimensión de salida igual a la entrada del bloque (32).

Normalización: estabilizan el aprendizaje

Dropout: con una tasa del 10%

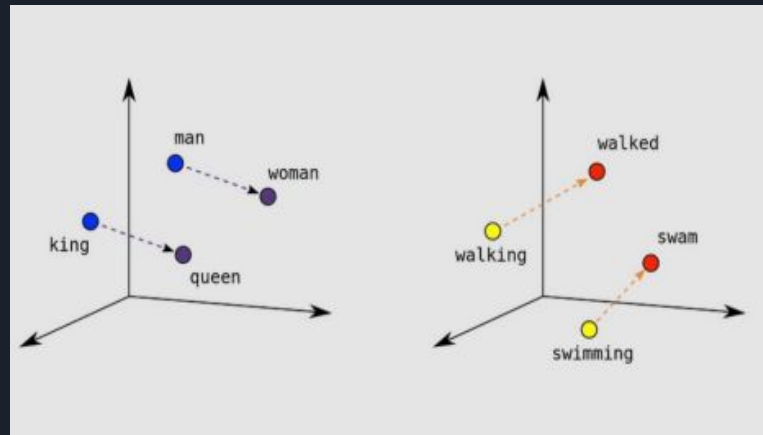


Capa de Embeddings

Embeddings de Tokens: Representan cada palabra en un espacio vectorial según su significado

Embeddings de Posiciones: Agregan información sobre el orden de las palabras.

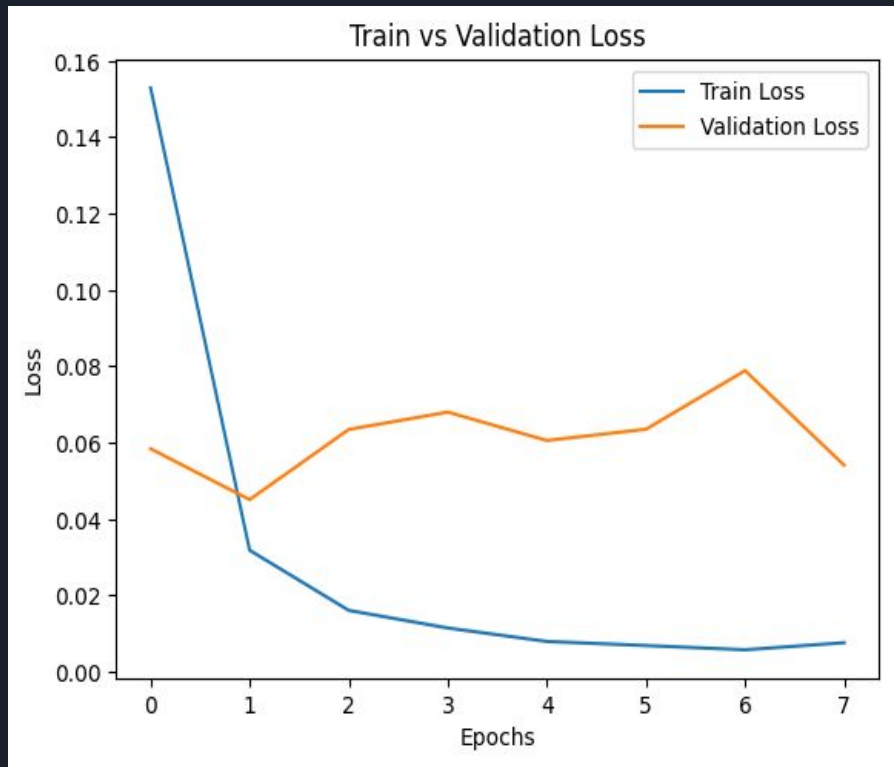
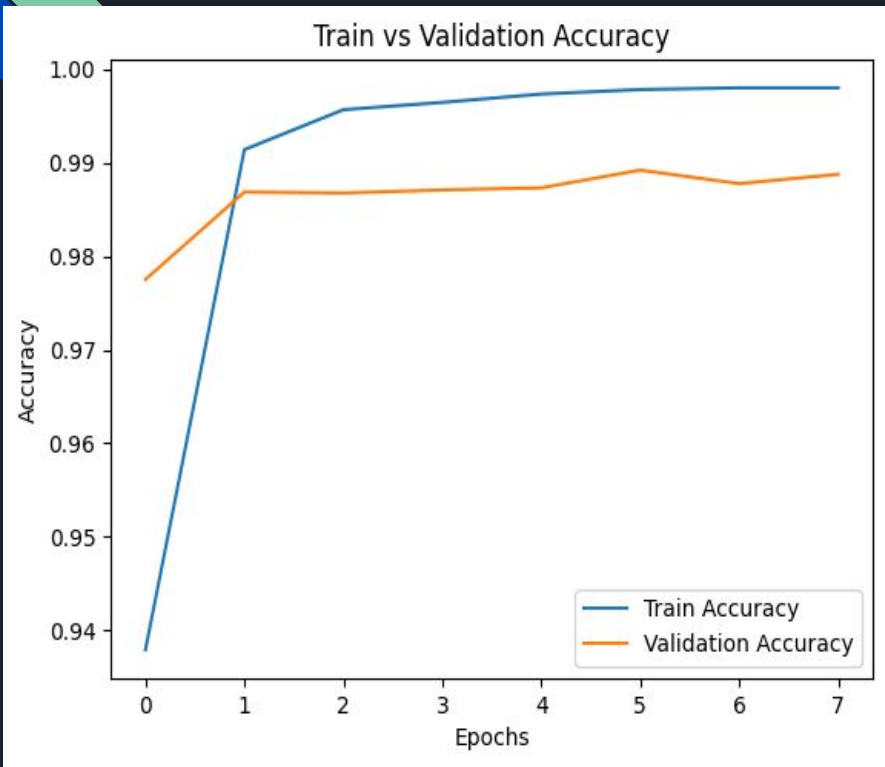
Esto permite que el modelo distinga entre palabras con el mismo significado pero en posiciones diferentes.




- Optimizador: Adam
- Pérdida: Binary Crossentropy
- Métricas: Accuracy

```
Epoch 1/50
842/842 ————— 111s 126ms/step - accuracy: 0.8602 - loss: 0.2873 - val_accuracy: 0.9798 - val_loss: 0.0601
Epoch 2/50
842/842 ————— 140s 124ms/step - accuracy: 0.9923 - loss: 0.0356 - val_accuracy: 0.9911 - val_loss: 0.0307
Epoch 3/50
842/842 ————— 140s 122ms/step - accuracy: 0.9977 - loss: 0.0136 - val_accuracy: 0.9888 - val_loss: 0.0342
Epoch 4/50
842/842 ————— 143s 123ms/step - accuracy: 0.9982 - loss: 0.0101 - val_accuracy: 0.9884 - val_loss: 0.0418
Epoch 5/50
842/842 ————— 103s 122ms/step - accuracy: 0.9990 - loss: 0.0071 - val_accuracy: 0.9846 - val_loss: 0.0638
Epoch 6/50
842/842 ————— 158s 142ms/step - accuracy: 0.9988 - loss: 0.0067 - val_accuracy: 0.9898 - val_loss: 0.0700
Epoch 7/50
842/842 ————— 129s 127ms/step - accuracy: 0.9994 - loss: 0.0040 - val_accuracy: 0.9893 - val_loss: 0.0655
Epoch 8/50
842/842 ————— 140s 124ms/step - accuracy: 0.9988 - loss: 0.0048 - val_accuracy: 0.9814 - val_loss: 0.1547
```

Se utilizó EarlyStopping para detener el entrenamiento si la pérdida de validación no mejora en 6 épocas.





Modelo	Accuracy
Logistic Regression	99,73%
Naive Bayes	96,5%
KNN	89,08%
Transformers	98,88%



Conclusiones

La **regresión Logística es el modelo más efectivo**, logrando una precisión del **99.73%** gracias a su capacidad para clasificar con alta exactitud. Los **Transformers**, con una precisión del **98.85%**, también muestran un **rendimiento sobresaliente**, siendo más robustos y adecuados para problemas complejos, aunque con un mayor costo computacional.

En contraste, **Naive Bayes (96.50%)** y **KNN (89.08%)** presentan un **desempeño** considerablemente **menor**, con limitaciones en la captura de relaciones complejas y un número significativo de errores en el caso de KNN.

Estos resultados subrayan la importancia de seleccionar modelos avanzados y bien ajustados según la naturaleza del problema y los recursos disponibles.